PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00, 9/445

A1

(11) International Publication Number: **WO 97/46932**

(43) International Publication Date: 11 December 1997 (11.12.97)

(21) International Application Number: PCT/US97/00724

(22) International Filing Date: 14 January 1997 (14.01.97)

(30) Priority Data:

08/657,889	7 June 1996 (07.06.96)	US
08/665,185	14 June 1996 (14.06.96)	US

(81) Designated States: JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

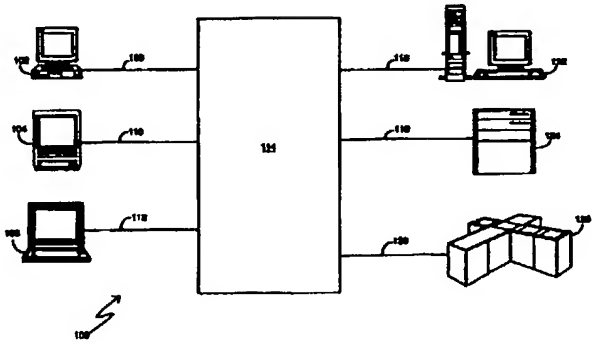
Published
With international search report.

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, Mail Stop 68, Sunnyvale, CA 94088-3453 (US).

(72) Inventors: LEE, Sherman; 28531 Cedarbluff Drive, Rancho Palos Verdes, CA 90275 (US). KYLE, David, G.; 3107 Barton Point Circle, Austin, TX 78733 (US).

(74) Agent: RODDY, Richard, J.; Advanced Micro Devices, Inc., One AMD Place, Mail Stop 68, Sunnyvale, CA 94088-3453 (US).

(54) Title: SYSTEM FOR MODIFYING COMPUTER RELATED SYSTEMS



(57) Abstract

A computer overall system (100) includes client systems (102, 104, 106) which are centered around customer computers. Client systems are connectable through a communications network (114) with server systems (122, 124, 126) by way of a LAN (108, 116): wan (110, 118); or POTS (112, 120). Communications network (114) can be any type of interconnect which is connected to respective server system. A computer which is part of a client system or a server system, includes therein a microprocessor that is programmed by the manufacturer with various items of information regarding the microprocessor itself. This programmed information defines a predetermined capability for the computer where its predetermined capability is defined by a set of parameters maintained within the computer system including several parameters such as the maximum core frequency (i.e. which is a measure of the speed or how many megahertz the central processing unit of the microprocessor is capable of running at), the maximum instruction set, a description of all the features available from the microprocessor, and other information related to the maximum capabilities of the microprocessor and the system it can operate, as appropriate for the specific application, for example, microprocessor may have parameters which broadly or narrowly defined the client system or the server system of which the microprocessor is a part. The microprocessor could also be provided with a hard coded electronic encryption key. Similarly, as is also known to those skilled in the art, the microprocessor could be configurable by changing internal software to change the core frequency or to modify yet other features.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

SYSTEM FOR MODIFYING COMPUTER RELATED SYSTEMS

Cross Reference to Related Applications

This application is a continuation of copending application USSN 08/657,889 filed June 7, 1996.

Field of the Invention

This invention relates to computer systems and, more particularly, to modifying computer related systems from a remote location.

Background of the Invention

Computer related systems contain many modifiable elements including hardware, firmware and software. Computer hardware includes, but is not limited to, single and multiple processors and microprocessors, which are herein individually and collectively referred to as microprocessors, peripheral devices, and individual processing units or portions thereof. Computer firmware includes, but is not limited to, programmable logic, read only memory, and other components in which control instructions such as microcode or program code can be mechanically, optically, electronically, or otherwise embedded. Computer software includes, but is not limited to, any form of information or instructions for the computer provided in any media including, but is not limited to, tape, disk, cards, read only memory (ROM), or random access memory (RAM).

In the past, hardware could only be modified where it was located. For example, between a personal computer and a peripheral printer, cable connections would have to be changed. Within a computer itself, various jumper cables would have to be changed or various components removed and replaced. For example, to upgrade the speed of a computer system, a microprocessor would have to be removed and replaced with a different, higher speed microprocessor.

Also in the past, since firmware is a combination of hardware with software embedded, it was similarly necessary to physically change components in order to modify or upgrade the computer system.

In the early days of this art, computer software originally started out with the same limitations as computer hardware and firmware in that physical replacement was required in order to modify the computer system; e.g., new software on a new media had to be "loaded," or read from the media and written, into the computer. Recently, however, software has been developed which contains a program which allows only certain features of the software to be utilized until a security code activation number or identification is entered which unlocks additional features.

This system of modifying computer software was a great step forward in that a replacement of the software media was not required. However, the supporting mechanisms for allowing these modifications to be made tended to be expensive, time consuming, error prone, and inefficient.

A computer customer wishing to modify this type of software would have to locate the telephone number of the software vendor, generally and unfortunately long after the software is

originally obtained and perhaps at a time when the original documentation had been misplaced or thrown-out. The customer would then contact the vendor, verbally give the vendor a serial number from the user manual or the computer screen, and then describe the configuration of the customer's computer system. The customer would then tell the provider what type of modification to the software was desired and receive a price quote. Upon agreeing to the price, the customer would provide credit card information to the provider and then verbally receive an activation code for the software. The customer would then have to use the activation code, for example, the customer would manually input it into the computer, to cause changes in the software.

As should be apparent, this software modification system is time consuming and laborious with the possibility of many errors being caused by the requirement for human intervention using numerical codes both on the customer's part as well as on the provider's part.

An additional problem with this system is that the software may be an unauthorized, or "bootleg," copy for which the user may not have obtained an appropriate right-to-use authority, or license, and for which the provider may not have received payment for the original software. Often, the basic software tends to be much more expensive than an upgrade.

Although there has been a long felt need to be able to automate the modification of computer systems by computer system to computer system contact through various communication networks, it has not been heretofore feasible because of the inability to protect both the customer and the provider of computer systems from those people who gain unauthorized access to the computer system. Such people who are unauthorized users of a computer are sometimes called computer "hackers." Computer hackers accessing the communications network between the two computer systems would be able to obtain access to confidential information of both systems and to obtain sensitive credit information from one and sensitive activation codes from the other. In addition, the hacker would be able to corrupt both systems by inserting viruses or changing activation codes or by causing or allowing other unfortuitous events to happen.

Thus, no system has been previously devised which is capable of controlling the modification of computer hardware, firmware, and/or software from a remote location despite the long felt need to do so.

Summary of the Invention

The present invention relates to a system in which a client computer system uses a predetermined activation code and includes the ability to access a server computer system through a communications network to receive modification commands based upon authentication of the activation code to cause modification of the server system.

The present invention provides an automated system for controlling other systems which include hardware, firmware, and software.

The present invention provides a system in which a client system may receive upgrade and

other information through a communications network from a server system which is able to confirm the identity of the client system.

The present invention further provides a mechanism for allowing a server system to determine the capability of a customer system and compatibility of upgrades or differing configurations for use with the customer system.

The present invention still further provides a mechanism for the elimination of human error in the modification of hardware, firmware, and software.

The present invention further provides a client system which is uniquely identifiable.

The present invention still further provides a client system which will accept software only from an authorized server system.

The present invention still further provides a server system which can uniquely identify a client system.

The present invention still further provides a server system which will provide software only to an authorized client system.

The above and additional advantages of the present system will become apparent to those skilled in the art from a reading of the following detailed description when taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a block diagram of an illustrative system embodying the present invention;

Fig. 2 is a block diagram of a client system that can be embodied in the system shown in Fig. 1;

Fig. 3 is a block diagram of a server system that can be embodied in the system shown in Fig. 1;

Fig. 4 is a block diagram of a personal computer that can be embodied in the client system of Fig. 1 or in the server system of Fig. 1;

Fig. 5 is a partial flow chart of a process that is usable in the client system of Fig. 2;

Fig. 6 is another partial flow chart which shows the remainder of the process that is partially shown in Fig. 5 and that is usable in the client system of Fig. 2;

Fig. 7 is a partial flow chart of a process that is usable in the server system of Fig. 3; and

Fig. 8 is another partial flow chart which shows the remainder of the process that is partially shown in Fig. 7 and that is usable in the server system of Fig. 3.

Detailed Description

COMPUTER SYSTEM 100

Referring now to Fig. 1, therein is shown an overall system 100 including client systems 102, 104, and 106 which illustrate a few examples of what are called client systems in what follows. There can in fact be any of number of different types of client systems. Client systems 102, 104, 106

that are shown here include systems which are centered around customer computers such as personal assistants, palmtops, laptop computers, personal computers, workstations, mini computers, mainframe computers, or massively parallel computers. Fig. 1 illustrates personal computer 102, workstation 104 and laptop computer 106, but could just as well include any of the mentioned type computers or other computers of that genre.

The client system 102 is shown connected to a local area network (LAN) 108, an example of which is an Ethernet system. LANs including Ethernet systems are well know in the art.

The client system 104 is shown connected to a wide area network (WAN) 110, which is exemplified by a T1 or an ISDN connection. T1 and ISDN systems are also well know in the art.

Client server 106 is shown connected to a plain old telephone system (POTS) 112, which is exemplified by a twisted pair of wires that are connectable between a network, like network 114, and a modem in a client system that may be located at a client's or other's site, like client system 106.

Client systems 102, 104, 106 are connectable by way of LAN 108, WAN 110, POTS 112 through communications network 114 with server systems 122, 124 126 by way of LAN 116; WAN 118; or POTS 120 respectively.

Communications network 114 can be any type of interconnect where it is not necessary for the inputs or outputs to be known, just that they be interconnectable. Network 114 can be a strictly customer internal network, a standard telephone system, a high data transmission network, among other structures such as a cable system or a combination of network systems. In the illustrative embodiment it could be the standard telephone system which includes capability to interconnect though what has become known in the art as the Internet.

The communications network 114 is connected by LAN 116, WAN 118, and POTS 120 respectively to server systems 122, 124, and 126. In that manner any of client systems 102, 104, 106 can communicate with any of server systems 122, 124, 126. It should also be noted that the communications network 114 as well as the various LAN 108, 116; WAN 110,118; and POTS 112, 120 can be either wired or wireless systems. Wireless systems could include, but not be limited to, radio frequency and infrared systems.

The server systems like server systems 122, 124, and 126 are systems typically operated or provided by hardware, firmware, and/or software vendors.

CLIENT SYSTEM 200, 102, 104, 106

Referring now to Fig. 2, therein is shown a typical client system 200 containing a computer 202. Client system 200 of Fig. 2 could be substituted as any of client systems 102, 104 or 106 of Fig. 1 and is here described as though it is any of those client systems. The computer 202 could be any type of computer including a personal assistant, palmtop, laptop, personal, workstation, mini, mainframe, or massively parallel computers, but in the illustrative embodiment described here is a personal computer.

The computer 202 functions, among other things, as a controller and is operatively connectable to peripheral devices in the client system 200 such as a printer 204, monitor 206, exterior memory 208, and miscellaneous other devices 210. The peripheral devices in addition could be smaller computers with microprocessors which have remote controllable deactivation and activation features to make them "green" peripheral devices which save power when they are not needed. Such devices are sometimes called "green" systems in the art meaning when they are in an "off" state to save power they are referred to as "red" whereas when they are in an "on" state and using power they are referred to as "green" - much in the same sense as a traffic signal used on highways to direct automobile traffic. This feature requires direct or indirect microprocessor to microprocessor communication such that the microprocessor in the computer 202 is capable of modifying the hardware, firmware, and software in the peripheral devices 204, 206, 208, 210.

The computer 202 is connectable by a communications interface 212 to the LAN 108, WAN 110 or POTS 112, but for purposes of this illustrative description are connected to LAN 108.

SERVER SYSTEM 300, 122, 124, 126

Referring now to Fig. 3, therein is shown the server system 300 having a computer 302 which could again be any type of computer such as a personal assistant, palmtop, laptop, desktop or personal, workstation, mini, mainframe, or massively parallel computer. Server system 300 of Fig. 3 could be substituted as any server systems 122, 124 or 126 of Fig. 1 and is here described as though it were any of those server systems and, in particular, in the illustrative embodiment of computer 302 within server system 300 is described as if it were a workstation.

The computer 302 functions, among other things, as a controller and is operatively connectable to a number of different additional devices and systems such as the database 304, phone system 306, or other miscellaneous systems 308. The database 304 could contain such information as various serial numbers that exist in systems, available features, configurations, prices, credit information, and activation codes. The phone system 306 could be a backup system to handle unusual situations. The miscellaneous devices 308 could include such systems as tracking systems to follow customer purchases and billing.

The computer 302 could be connected by a communication interface 310 to any of the LAN 116, WAN 118 or POTS 120, but for purposes of this illustrative description are connected to LAN 116 and thence to the communications network 114.

COMPUTER 400

Referring now to Fig. 4, therein is shown the computer 400 which could be of a type to embody computer 202 of Fig. 2 or computer 302 of Fig. 3 as a part, respectively, of client system 200, 102, 104, 106 or server system 300, 122, 124, 126. Computer 400 includes therein a microprocessor 402 that operates as a controller among other things. In particular, microprocessor 402 can be any type of microprocessor manufactured by a number of companies such as Advanced

Micro Devices, Intel Corporation, Cyrix, IBM, Digital Equipment Corporation, and/or Motorola among other companies.

The microprocessor 402 will have programmed in it various items of information regarding the microprocessor itself. Typically, the programming of microprocessor 402 is done by the manufacturer. This programmed information defines a predetermined capability for the computer where its predetermined capability is defined by a set of parameters maintained within the computer system including several parameters relating to the capabilities of the computer system such as the maximum core frequency (i.e., which is a measure of the speed or how many megahertz the central processing unit of the microprocessor is capable of running at), the (maximum) instruction set, a description of all the features available from the microprocessor, and other information related to the (maximum or other) capabilities of the microprocessor and the system in which it operates, as appropriate for the specific application, for example, microprocessor 402 may have parameters which broadly or narrowly defined the client system or the server system of which the microprocessor is a part. The microprocessor 402 has programmed within itself certain parameters such as its full identification including the type of central processing unit it has, the manufacturer, the class of the machine, and a unique serial number as well as perhaps still other relevant parameters which have meaning within the context of the application(s) that are associated with, or in whatever manner use, the microprocessor 402.

The microprocessor 402 could also be provided with a hard coded electronic encryption key. There are several types of encryption keys and methods of hard coding them that are well known to those skilled in the art. Similarly, as is also known to those skilled in the art, the microprocessor 402 could be configurable by changing internal software to change the core frequency or to modify yet other of its features or, for that matter, the features of the computer 400 of which microprocessor 402 is illustrated as a part.

In the illustrative embodiment that is being described, the microprocessor 402 is connected through additional memory 404 which is shown as RAM 404 but could be other forms of memory to serve a specific application. In advanced microprocessors, the additional memory would more likely be a L2 cache and, while it may be integral with the microprocessor 402, in the illustrative embodiment, it is shown as a separate chip. The L2 cache memory generally tends to be a high speed RAM memory.

Similarly, the microprocessor 402 is connected to a ROM 406, which is illustratively shown as an electrically erasable read only memory (EPROM) but could be other forms of memory to serve a specific alternative application. Again, the ROM 406 can be either separate or integral with the microprocessor 402 but, in the illustrative embodiment, it is shown as a separate chip

In our illustrative embodiment, the ROM 406 could be a separate component which could be programmed, as would be evident to those skilled in the art, with a serial number, feature control

information (as will hereinafter be described), a second encryption code, cyclic redundancy check code (to permit multiple checks on the accuracy of the previous code), and error correction code (to correct any errors in the previous code). Again as would be evident to those skilled in the art, upon a comprehension of the principles of the instant invention and of the description herein, the second encryption code key could be used to code and decode the serial number and feature information.

In the illustrative embodiment described here, the microprocessor 402, the RAM 404 and the ROM 406 are all mounted on a card 408, called a "daughter" card 408 in the art, which is insertable into a system planar (not shown) in the computer 400.

CLIENT UPGRADE PROCEDURE 500

Referring now to Figs. 5 and 6, therein are shown a process in flow chart format that could be embodied in software as a computer program which are typical of the ones resident in the client systems 102, 104, 106, 200. The illustrated process is herein referred to as the client upgrade procedure 500 in that the process is useful for upgrading a client system in order to modify it based on payment of an appropriate agreed upon price and being provided an appropriate verification that authenticates the upgrade. The upgrade procedure 500 can be embodied in software.

The process 500 starts in Fig. 5 at a "upgrade procedure" block 502 that is entered in response to an upgrade request, typically from a client, and proceeds to "connect to upgrade server" block 504. Then the program proceeds to the "is connection successful?" decision block 506. If no, the program proceeds to "retry connection?" decision block 508 and ends at "end" block 512 when a sufficient number of retries has been unsuccessful attempted or, if yes, iterates through "connect to upgrade server" block 504 to retry a connection.

If the connection is successful out of decision block 506, the program proceeds to the "transmit product information for authentication" block 510 and then proceeds to the "is authentication successful?" decision block 514. If not successful, the program goes to the "initiate authentication failure procedure" block 516 and then through connector blocks 532 and 620 on to "restore system to original state" block 618, shown in Fig. 6.

If the authentication is successful, the program proceeds to the "transmit upgrade request to server" block 518. The program then proceeds to read "is upgrade request acknowledged?" decision block 520. If no, the program goes to the "inform customer that upgrade request is not valid" block 522 and then through connector blocks 532 and 620 on to "restore system to original state" block 618.

If the upgrade request is acknowledged, the program proceeds to the "receive upgrade cost from server" block 524. Upon receipt of the upgrade cost from the server, the program then goes to the "is upgrade cost acceptable?" decision block 526. If the upgrade cost is not acceptable, the program goes to the "cancel upgrade" block 528 and then through connector blocks 532 and 620 on to "restore system to original state" block 618.

Generally, in the illustrative embodiment the "initiate authentication failure procedure" block 516 and the "inform customer that upgrade request is not valid" block 522 end at the "restore system to original state" block 618. While block 618 could restore the original system state and proceed to disconnect the connection, it should also be understood that there may be a further program or subroutine to try to take corrective action or provide additional information. The design of these could be within the ability of those having ordinary skill in the art. The "restore system to original state" block 618 is also the termination point of the "initiate forgery procedure" block 614 and eventually through blocks 624 and 628 proceeds to the "terminate connection to update server" block 630.

If the upgrade cost is acceptable, the program will go to the "receive upgrade applet from server" block 604 through connector blocks 530 and 602. An "Applet" is a small application program which is capable of running on the server system to upgrade hardware, firmware, or software in the client system.

The program procedure 500 then proceeds to the "is upgrade applet valid?" decision block 608. If the upgrade applet is not valid, the program proceeds to the "is upgrade applet a forgery?" decision block 610. If the applet is not a forgery, the program goes to the "request retransmission upgrade applet" block 606 and iterates by returning to the "receive upgrade applet from server block 604." If the upgrade applet is a forgery, the program proceeds to the "initiate forgery procedure" block 614 and through block 618 eventually proceeds to the "terminate connection to update server" block 630.

If the upgrade applet is valid, the program proceeds to the "execute upgrade applet to perform requested upgrade" block 612. After the execution, the program proceeds to the "is upgrade successful?" decision block 616.

If the upgrade is successful out of decision block 616, the program proceeds to the "transmit successful upgrade message to server" block 622, then to the "inform end-user of successful upgrade" block 626, and then to the "terminate connection to upgrade server" block 630.

If the upgrade is unsuccessful out of the decision block 616, the program proceeds to the "restore system to original state" block 618 and through blocks 624 and 628 eventually proceeds to the "terminate connection to update server" block 630, as is now described. When the system is restored to the original state, the program proceeds to the "transmit upgrade error message to server" block 624 and then to the "inform customer of upgrade error" block 628. From the block 628 the program proceeds to the "terminate connection to upgrade server" block 630.

As part of the successful upgrade, the "terminate connection to upgrade server" block 630 will cause the program to proceed to the "request system restart" block 632 to cause the system to reboot to reinitialize the system with the upgraded system parameters.

The program for the client system upgrade procedure 500 then ends at "end" block 634.

SERVER UPGRADE PROCEDURE 700

Referring now to Figs. 7 and 8, therein are shown the partial flow charts of the typical computer program which is resident in the server systems 122, 124, 126, 300. The illustrated process is herein referred to as the server upgrade procedure 700 in that the process is useful for illustrating what the server system does when a client seeks an upgrade, i. e. helps show the interplay between client system and server system as the client seeks an upgrade.

The server upgrade procedure 700 starts at the "upgrade procedure" block 702 in Fig. 7 in response to a client's upgrade request.

From block 702, the program goes to the "connection from client?" decision block 704. If there is no connection, the program iterates through block 704 and continues to check for the connection. If there is connection, the program proceeds to the "receive product information for authentication" block 706 and then the program proceeds to the "is product information valid?" decision block 708.

If the product information is not valid, the program proceeds to "request retransmission of product information" block 710 which returns the program to the block 706 for another iteration through block 708.

If the product information is valid, the program proceeds to the "authenticate product information" block 712 and then proceeds to the "is product information authenticated?" decision block 714. If the product information is not authenticated, the program goes to "transmit authentication invalid message" block 716 and through connector blocks 728 and 804 on to "terminate connection to client" block 830 in Fig. 8, which is also referred to herein as "disconnect" block 830 in Fig. 8.

If the product information is authenticated, the program proceeds to the "transmit authentication valid message" block 718. The program then proceeds to the "receive upgrade request" block 720 and then proceeds to the "is upgrade request valid?" decision block 722.

If the upgrade request is invalid, the program proceeds to the "transmit upgrade invalid message" block 724 and through connector blocks 728 and 804 on to block 830 in Fig. 8. Again, the "disconnect" block 830 may actually be another program.

If the upgrade request is valid, the program will proceed through connectors 726 and 802 to the "transmit upgrade valid message" block 806 shown in Fig. 8.

From the block 806, the program proceeds to the "access upgrade cost database" block 808 and then proceeds to the "transmit upgrade cost data" block 810. From block 810 the program waits to receive the upgrade cost acknowledge at block 812.

After the cost data is transmitted, the program goes to the "is upgrade cost acceptable?" decision block 814. If the cost is not accepted, the program will go to the "cancel upgrade request" block 816 and then to the "disconnect" block 830. Again, the "disconnect" block 830 may be an

additional program which contains advertising.

If the cost is accepted, the program will go to the "transmit upgrade applet" block 818 and then to the "is upgrade applet received correctly?" block 820. If the applet is not received correctly, the program will go to the "retransmit applet request" block 822 where if there is a retransmission request it will iterate back to the "transmit upgrade applet" block 818; else it will go to "disconnect" block 830.

If the applet is received correctly, the program will proceed to the "receive upgrade status message" block 824.

From the "receive upgrade status message" block 824, the program proceeds to the "is upgrade successful?" decision block 826 in Fig. 8.

If the upgrade is not successful, the program will go to the "disconnect" block 830 which could be a checking program to determine why the upgrade is unsuccessful.

If the upgrade is successful, the program will proceed to the "initiate upgrade charge procedure" block 828 which could be the procedure for billing and handling the receipt of compensation for the upgrade.

After the "initiate upgrade charge procedure" block 828, the program will proceed to the "disconnect" or "terminate connection to the client" block 830. After termination of the connection, the program will proceed to the "update product information database" block 832 and then will proceed to the "end" block 834.

SOME MORE OF THE INTERACTION BETWEEN
CLIENT UPGRADE PROCEDURE 500 AND SERVER UPGRADE PROCEDURE 700

In operation, a customer desirous of modifying the customer's client system could initiate the client upgrade procedure 500, which could be embodied in computer 400, as a computer program using the flow chart shown in Figs. 5 and 6. Computer 400 in turn can be embodied in computer 202 of client system 200, 102, 104, 106.

Although different types of modifications of the client system are possible, for example, the reduction of capabilities in order to reduce licensing costs (e.g. to reduce the costs to use the software that is being modified using the principles of our invention) or just changing a configuration to specialize client system 102 in a particular manner, the following discussion is that of an illustrative embodiment that is related to upgrades.

When the customer is interested in an upgrade, the customer could initiate the upgrade procedure 500 at entry point 502 in any of the client systems 200, 102, 104, 106 to start the program embodying the client upgrade procedure 500. The program could then, without further customer intervention, look up the necessary connective path through the communications network 114 to the appropriate server system 300, 122, 124, 126. This path connection (504) could be done by having the program have the telephone number or the electronic address of the appropriate server system

300, 122, 124, 126.

The program could then check to determine whether the connection is successful in the "is connection successful" block 506 and if not successful, it could retry the connection in the "retry connection?" decision block 508 a predetermined or an adjustable number of retries and, if unsuccessful within that number of retries, it will proceed to the procedure end block 512.

On the other hand, if the connection is successful within the aforementioned number of retries, the microprocessor 402 can transmit (510) the encrypted information relating to serial numbers, featured capabilities, etc., across the communications network 114 to the appropriate server system 300, 122, 124, 126 in which the server upgrade procedure 700 of Figs. 7 and 8 can also be embodied in software. Server upgrade process 700 begins to operate at upgrade procedure 702 and upon being connected to a client server (704) receives the information at the "receive product information for authentication" block 706.

The server upgrade procedure 700 then checks at the "is product information valid?" decision block 708 shown in Fig. 7 to determine if the serial numbers and related information match or are otherwise valid. It should be noted at this point that this type of information can be used to determine if various components have been stolen or are otherwise being used by unauthorized persons. Similarly, this information can be used for other purposes such as determining if recalls are required due to product liability problems.

The vendor system then authenticates the product information in the "authenticate product information" block 712 and the "is product information authenticated?" decision block 714.

If the product is authenticated, the vendor's system transmits a validation message at the "transmit authentication valid message" block 718 which is received by the client system 102 at the "is authentication successful?" decision block 514 in Fig. 5.

When the authentication is successful, the client system 102 transmits a signal at the "transmit upgrade request to server" block 518 through the communications network 114 to the server system here assume it to be server system 122 to be received at the "receive upgrade request" block 720 in Fig. 7. The server program then proceeds to determine if the upgrade request is valid at the "is upgrade request valid?" decision block 722. At this point, the previously provided message on the product information is used to check to make sure that the client system is capable of accepting the upgrades and performing them.

If the upgrade request is valid, the server system 122 proceeds to retrieve the cost information related to the upgrade at the "retrieve costs from cost database" block 808 in Fig. 8.

The server system 122 then transmits the cost data at "transmit cost data" block 812 through the communications network 114. The client system 102 at the "receive upgrade cost from upgrade server" block 524 in Fig. 5. The customer must then determine whether or not the upgrade cost is acceptable (526). If not, the customer can cancel the upgrade (528), but if the cost is acceptable, the

client system 102 will so indicate to the server system 122 to proceed with transmitting the upgrade applet at the "transmit upgrade applet" block 818 in Fig. 8.

The transmitted applet will be received at the client system 102 at the "receive upgrade applet from upgrade server" block 604 in Fig. 6. The upgrade applet could then be checked (608) to assure that it is correct, by any well known error correcting and checking procedure, or to determine whether or not it is a forgery (610) and an attempt is being made to harm the client system 102 by, for example, downloading a virus program. It should be kept in mind that the double encryption procedure of the computer 400 permits the encryption of information going in both directions and thus assures the protection of ingoing information as well as outgoing information.

The computer 400 then executes the upgrade applet at the "execute upgrade applet to perform requested upgrade" block 612.

Once the upgrade applet is performed, the client system 102 then checks to determine if the upgrade has been successful at the "is upgrade successful?" decision block 616.

If the upgrade is not successful, it is necessary to restore the system back to its original state so that any partial attempts at the upgrade which were incorporated will not later adversely affect a successful upgrade. And this is done at the "restore system to original state" block 618. The fact that there has been an error in the attempted upgrade will need to be transmitted back to the server system 122 as well as to a customer and this is done, respectively at the "transmit upgrade error message to upgrade server" block 624 and the "inform end-user of upgrade error" block 628.

If the upgrade is successful, as determined by a conventional capability checking program, the client system 102 will transmit a message at the "transmit successful upgrade message to upgrade server" block 622 which will be received at the server system 122 at the "is upgrade successful?" block 826 in Fig. 8.

When the upgrade is successful, the server system 122 will initiate the charging procedures to cover the cost of the upgrade at the "initiate upgrade charge procedure" 828. It should be evident that there are a number of ways that the charge procedure can operate. One example could be the use of credit card information transmitted via the double encryption procedure in the computer 400.

The fact that the various encryption keys can be changed at will and randomly and totally without human intervention means that the breakdown of one pair of encryption keys will not compromise any other set of encryption keys which makes the system extremely secure for the transmittal of financial information or programming without the exterior introduction of computer viruses.

As part of the upgrade charge procedure, the server system 122 could advise the client system and the customer of the successful upgrade and payment. The server system 122 could then terminate the connection to the client system at the "terminate connection to the client" block 830 and then could proceed to the "update product information database" block 832 which could allow the

collection of marketing data and other information related to the particular product.

After the termination of the connection to the server system 122 at the "terminate connection to upgrade server" block 630 in Fig. 6, the program will proceed to the "request system restart" block 632 to reboot the system to initialize all the initialize parameters to be consistent with the upgrade which has been provided.

It should be noted that there are a number of advantageous features of the present invention which have been described peripherally. For example, this system is capable of tracking thefts of individual components in a computer system, preventing forged software from entering the system, and providing secure transmittal of financial information. While the invention has been described in conjunction with a specific embodiment, it is to be understood that many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, it is intended to embrace all such alternatives, modifications, and variations which fall within the spirit and scope of the appended claims.

We claim:

1. Apparatus 100, connectable to a communications network 114, comprising:

a first computer system 200, 102, 104, 106 connectable to said communications network 144 and having a predetermined capability, the predetermined capability being defined by a set of parameters maintained within the first computer system; said first computer system including control means 202, 402 for controlling said predetermined capability; and

a second computer system 300, 122, 124, 126 connectable to said communications network; said second computer system including control means 302, 402 for identifying and controlling said first computer system control means through said communications network.

2. The apparatus as claimed in claim 1 wherein said first computer system includes identifier means for uniquely identifying said first computer system.

3. The apparatus as claimed in claim 2 wherein said second computer system includes authentication means for identifying said identifier means.

4. The apparatus as claimed in claim 3 wherein said first computer system includes encryption means for encrypting said identifier means whereby said unique identification of said first computer system is encrypted; and

said second computer system includes decryption means for decrypting said encryption means encryption of said identifier means whereby said unique identification of said first computer system is accomplished across said communications network.

5. The apparatus as claimed in claim 4 wherein said first computer system includes further encryption means for further encrypting said encryption means encryption of said identifier means; and

said second computer system includes further decryption means for further decrypting said further encrypting means of said first computer system whereby said unique identification of said first computer system has at least two levels of encryption.--;.

6. Apparatus 100, connectable through a communications network 114 to a server computer system 300, 122, 124, 126 having a control mechanism 302, 402 responsive to signals through said communications network 114, comprising:

an identifiable computer system 200, 102, 104, 106 connectable to said communications network 114;

said identifiable computer system having a predetermined capability;

said identifiable computer system including control means 202, 402 for controlling said predetermined capabilities; and

said identifiable computer system including signaling means for sending a signal through said communications network to said server computer to cause said server computer to control said identifiable computer system through said communications network.

7. The apparatus as claimed in claim 6 wherein said identifiable computer system includes identifier means for uniquely identifying said identifiable computer system.

8. The apparatus as claimed in claim 7 including means for transmitting said identifier means unique identification of said identifiable computer system.

9. The apparatus as claimed in claim 8 includes encryption means for encrypting said transmission means transmission of said identifier means.

10. The apparatus as claimed in claim 9 wherein said identifiable computer system includes further encryption means for further encrypting said encryption means transmission of said identifier means.

11. Apparatus 100, connectable through a communications network 114 to a client computer system 200, 122, 124, 126 having a predetermined capability and a control mechanism 202, 402 for sending a signal through said communications network to request modification to certain of said predetermined capability and responsive to a signal through said communications network 114 to control said predetermined capability, comprising:

a computer system 300, 122, 124, 126 connectable to said communications network 114;

said computer system having memory means 304 containing information regarding said predetermined capability in said client computer system; and

said computer system responsive to said signal from said client computer requesting modification of said predetermined capability to obtain one of said combinations of said predetermined capability to provide a control signal through said communications network for said client computer to control said predetermined capability in said client computer.

12. The apparatus as claimed in claim 11 wherein said computer system includes identifier means for uniquely identifying said control mechanism of said client computer.

13. The apparatus as claimed in claim 12 wherein said computer system includes authentication means for identifying said signal from said control mechanism of said client computer.

14. The apparatus as claimed in claim 13 wherein said computer system includes decryption means for decrypting said signal from said control mechanism of said client computer.

15. The apparatus as claimed in claim 14 wherein said computer system includes further decryption means for further decrypting said decryption means.

16. Apparatus 100, connectable to a communications network 114, comprising:

a first computer system 200, 102, 104, 106 connectable to said communications network 114 and having a predetermined set of capabilities; said first computer system including control means 202, 402 in controlling any of said capabilities in said predetermined set of capabilities; and

a second computer system 300, 122, 124, 126 connectable to said communications network 114; said second computer system including control means 302, 402 for controlling said first computer system control means through said communications network 114.

17. Apparatus 100, connectable through a communications network 114 to a server computer system 300, 122, 124, 126 having a control mechanism responsive to signals through said communications network, comprising:

a client computer system 200, 102, 104, 106 connectable to said communications network 114;

said client computer system having a predetermined set of capabilities;

said client computer system including control means 202, 402 for controlling any of said capabilities in said predetermined set of capabilities; and

said client computer system including signaling means for sending a signal through said communications network to said server computer to cause said server computer to control said client computer system through said communications network.

18. Apparatus 100, connectable through a communications network 114 to a client computer system 200, 102, 104, 106 having a predetermined set of capabilities and a control mechanism for sending a signal through said communications network to request modification to certain of said predetermined set of capabilities and responsive to a signal through said communications network to control said predetermined set of capabilities, comprising:

a server computer system 300, 122, 124, 126 connectable to said communications network 114;

said server computer system having memory means 304, 404, 406 containing information regarding combinations of said predetermined set of capabilities in said client computer system; and

said server computer system responsive to said signal from said client computer requesting modification of said predetermined set of capabilities to obtain one of said combinations of said predetermined set of capabilities from said memory means to provide a control signal through said communications network for said computer system to control said predetermined set of capabilities in said client computer.

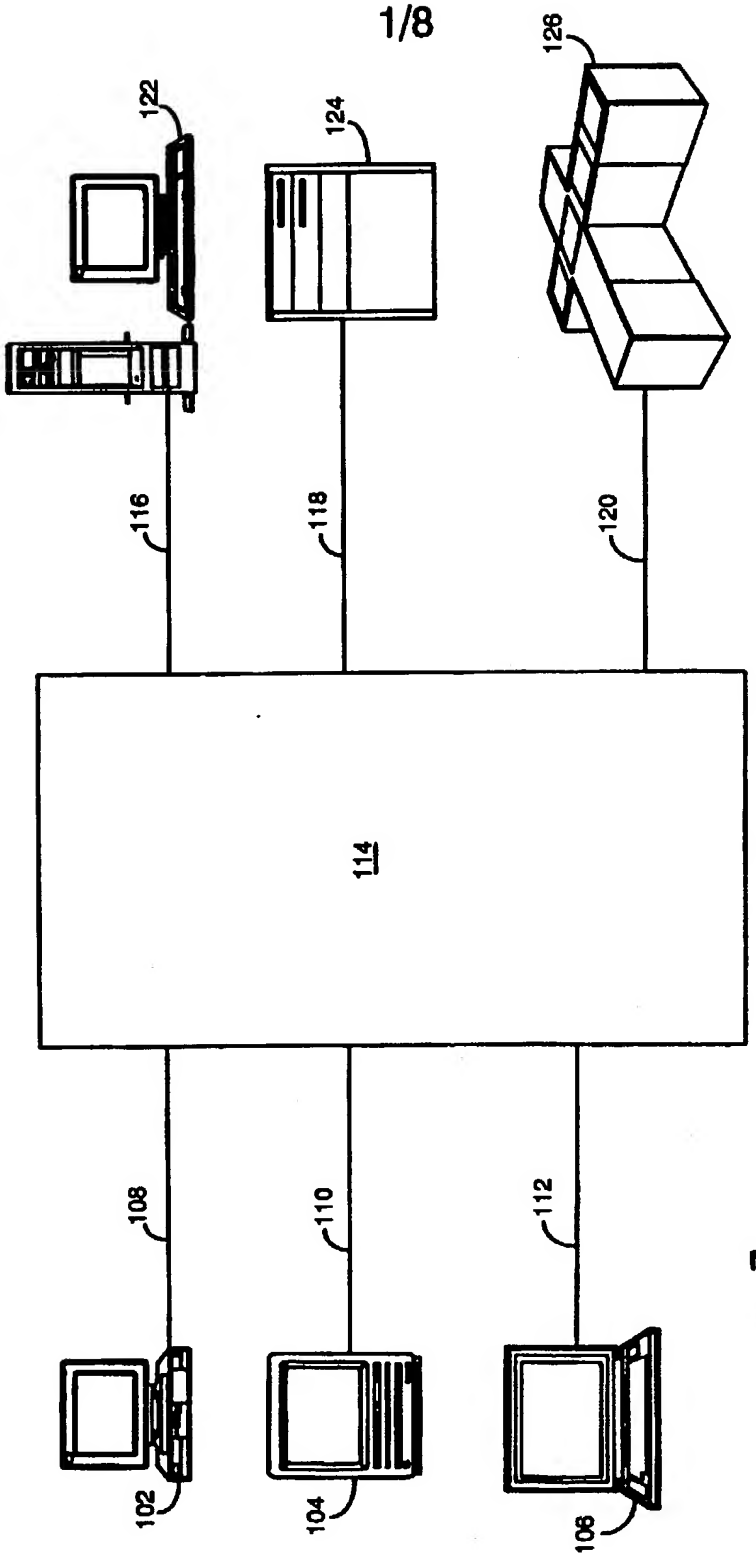


FIG. 1

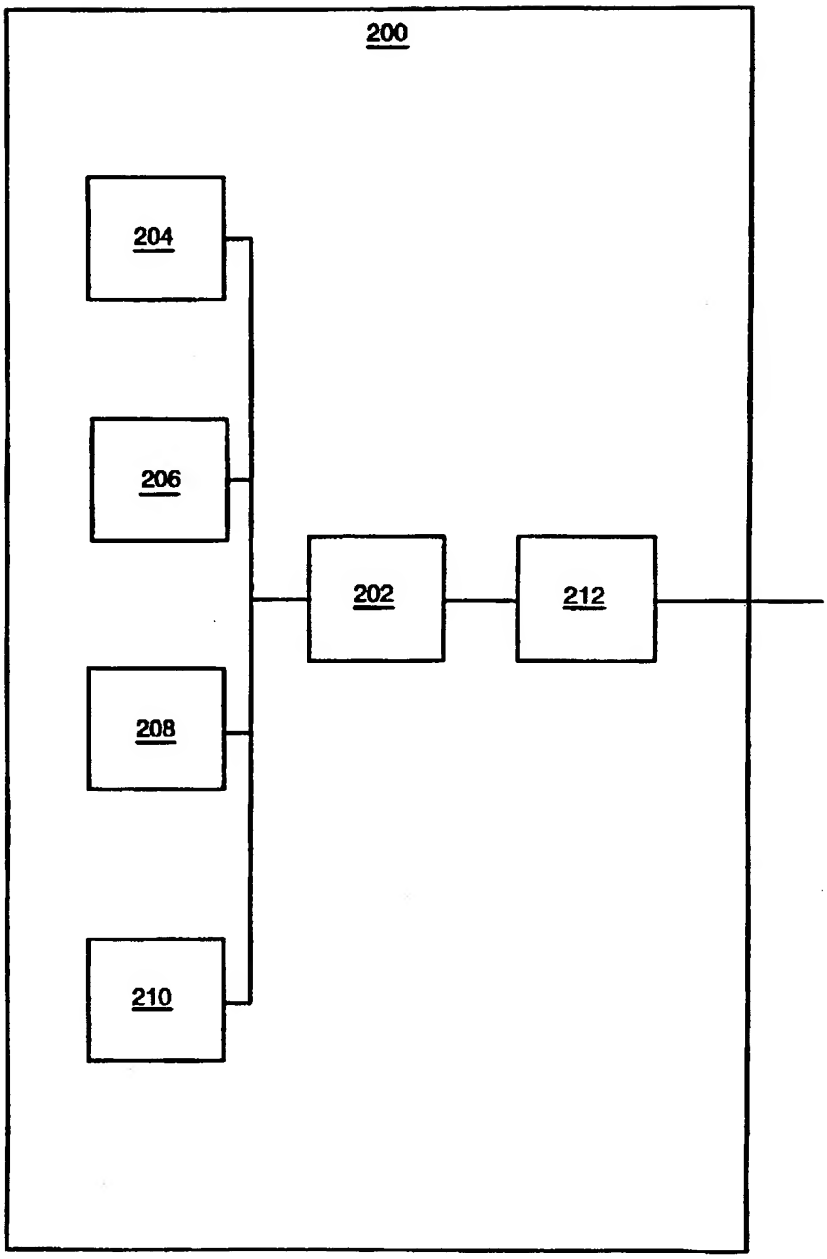


FIG. 2

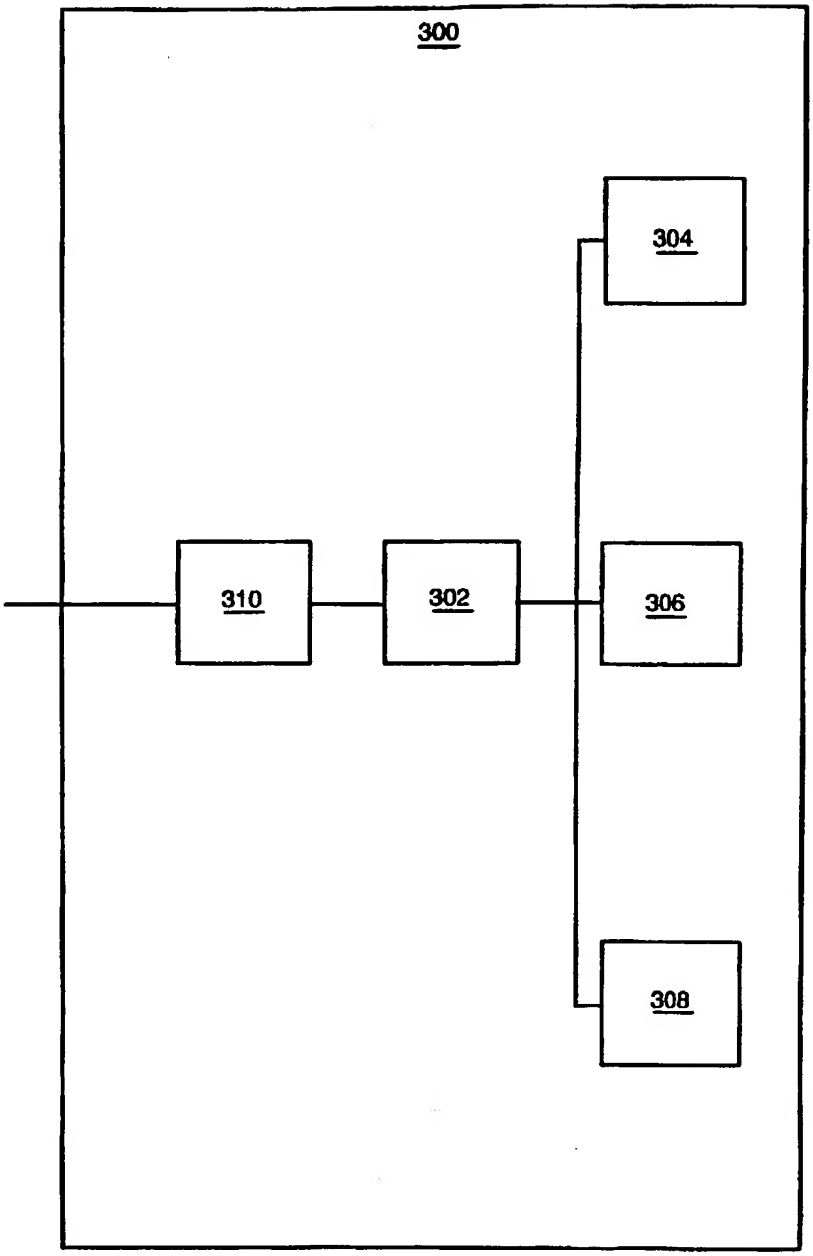


FIG. 3

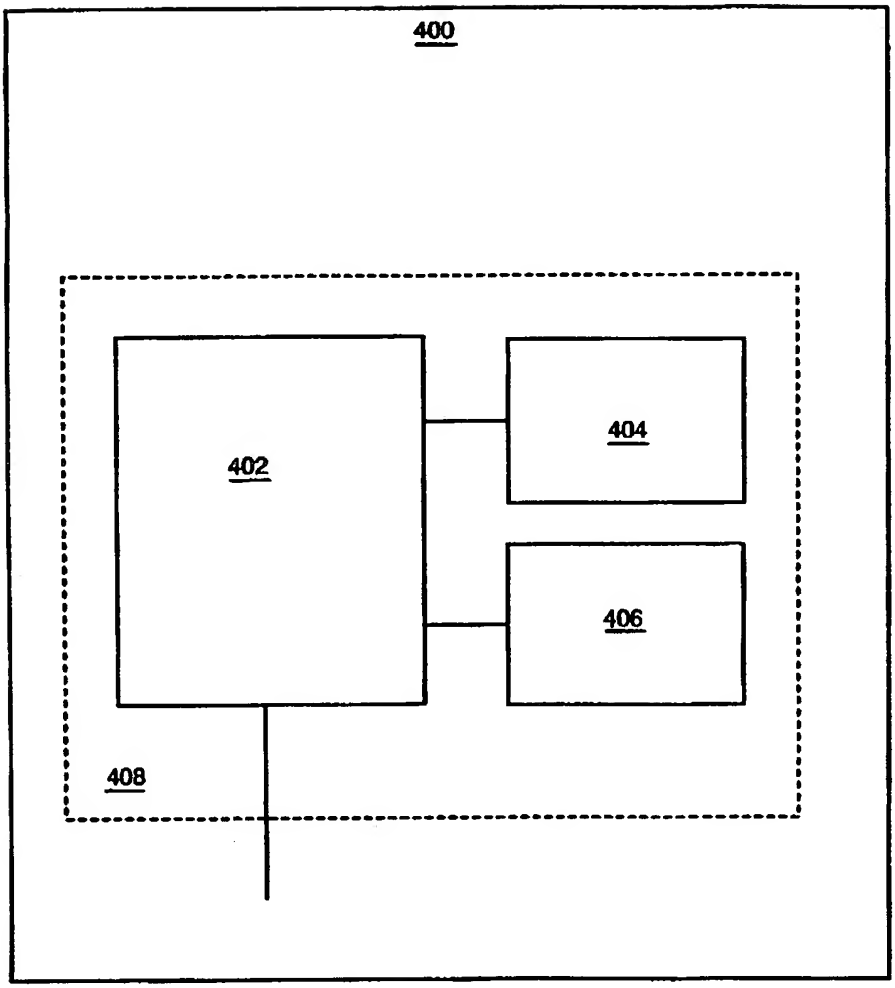


FIG. 4

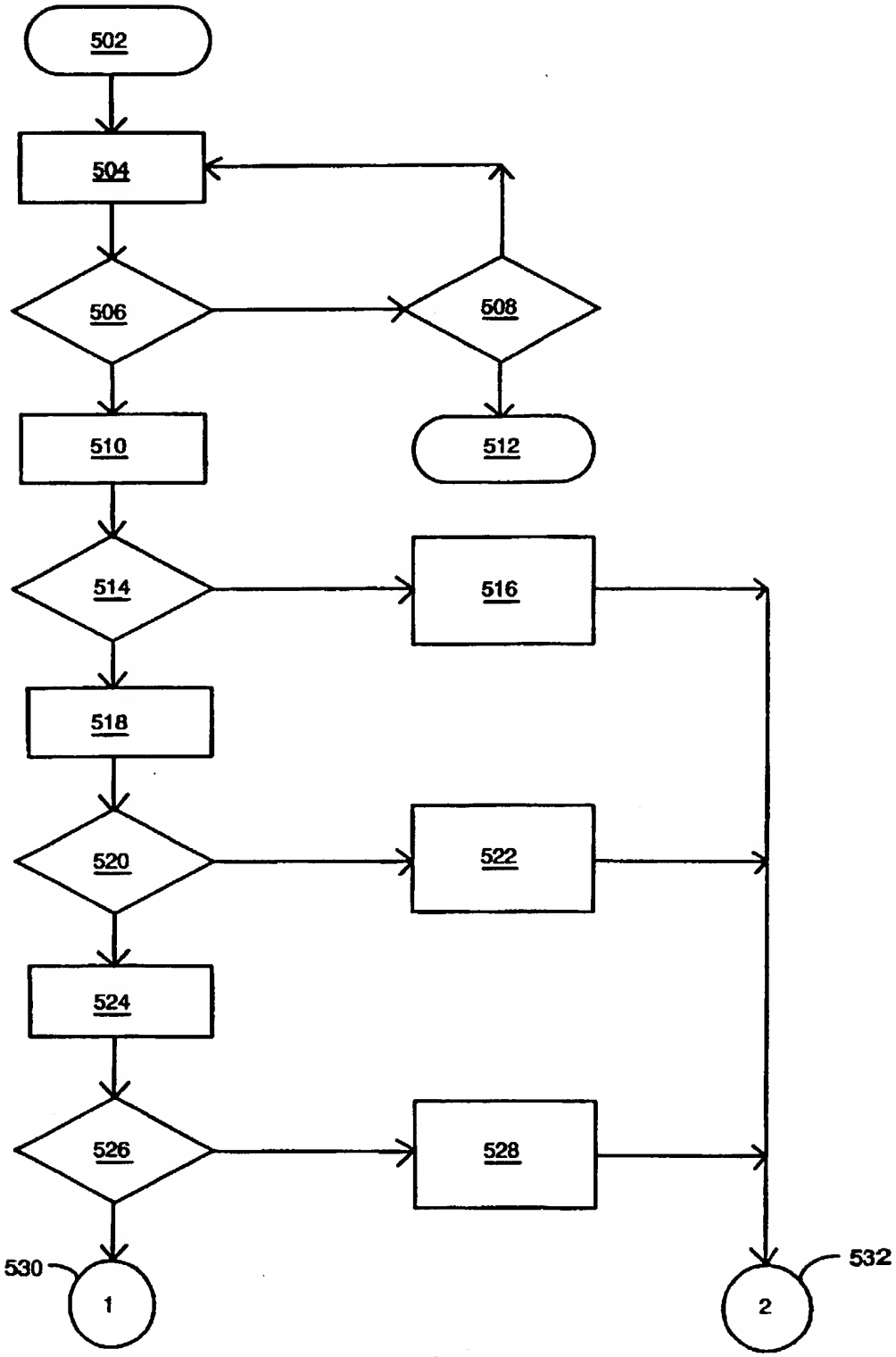

502
504
506
508
510
512
514
516
518
520
522
524
526
528
530
1
532
2

FIG. 5

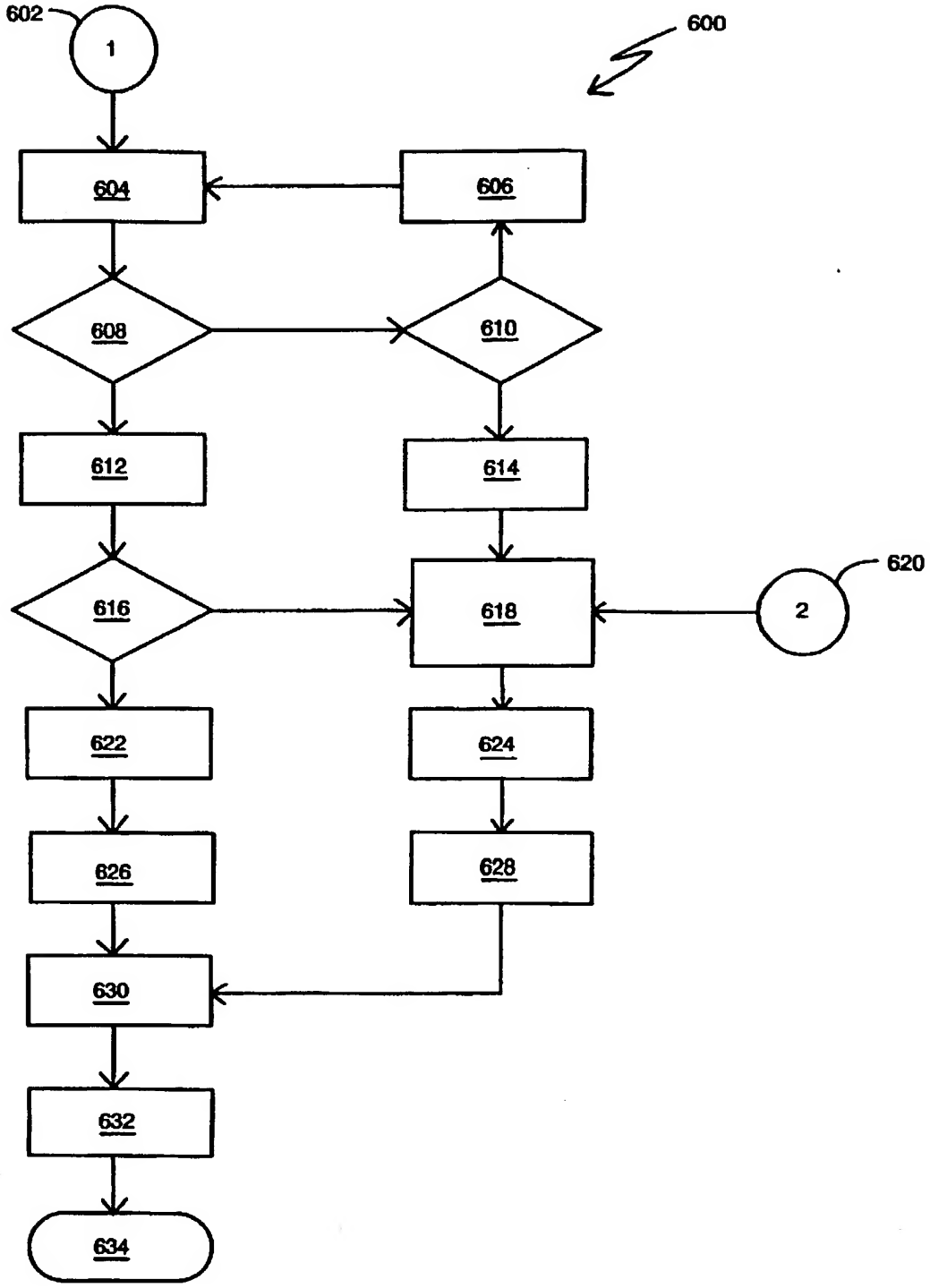

602
1
600
604
606
608
610
612
614
616
618
620
2
622
624
626
628
630
632
634

FIG. 6

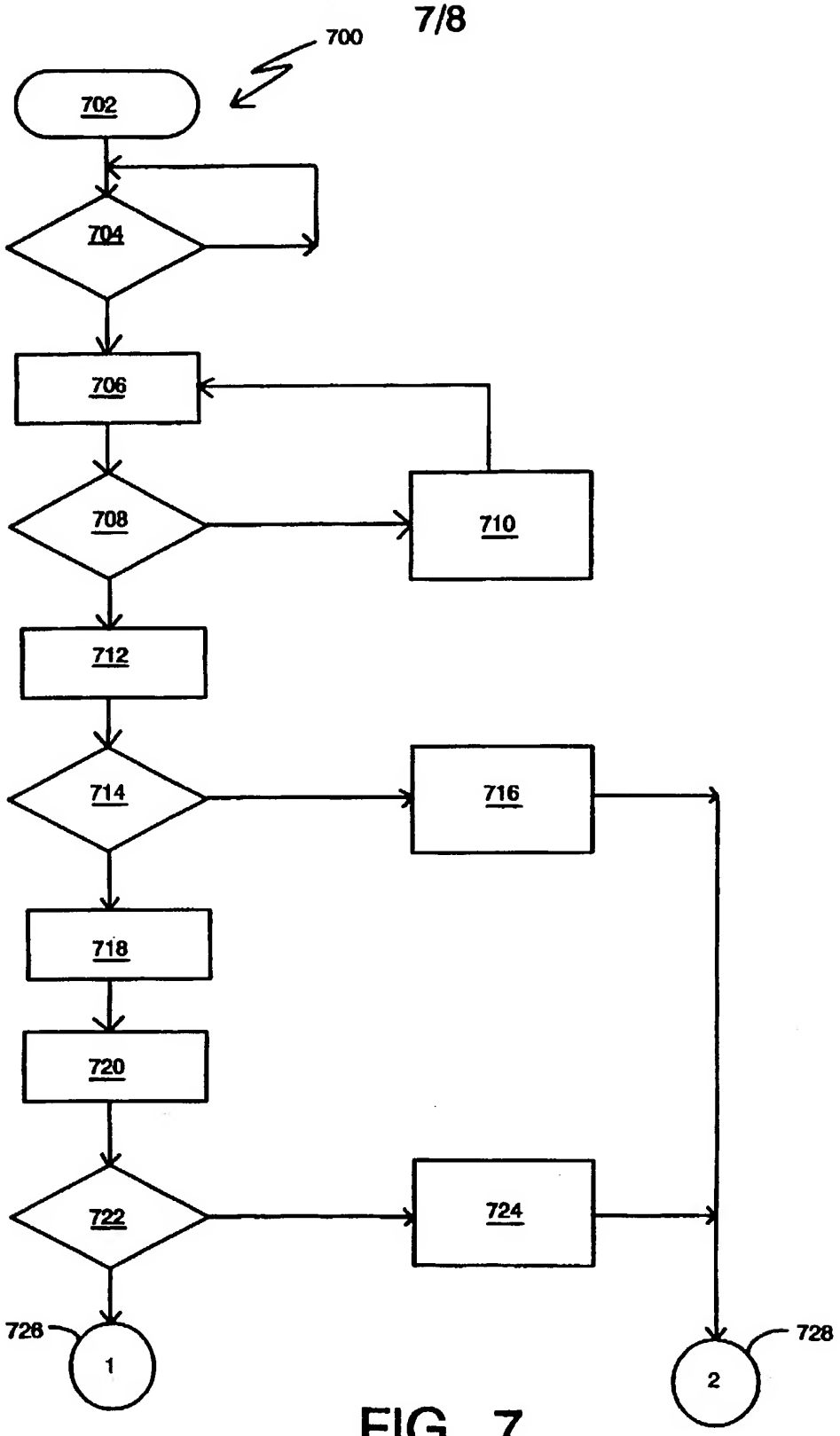

700
702
704
706
708
710
712
714
716
718
720
722
724
726
1
728
2
728

FIG. 7

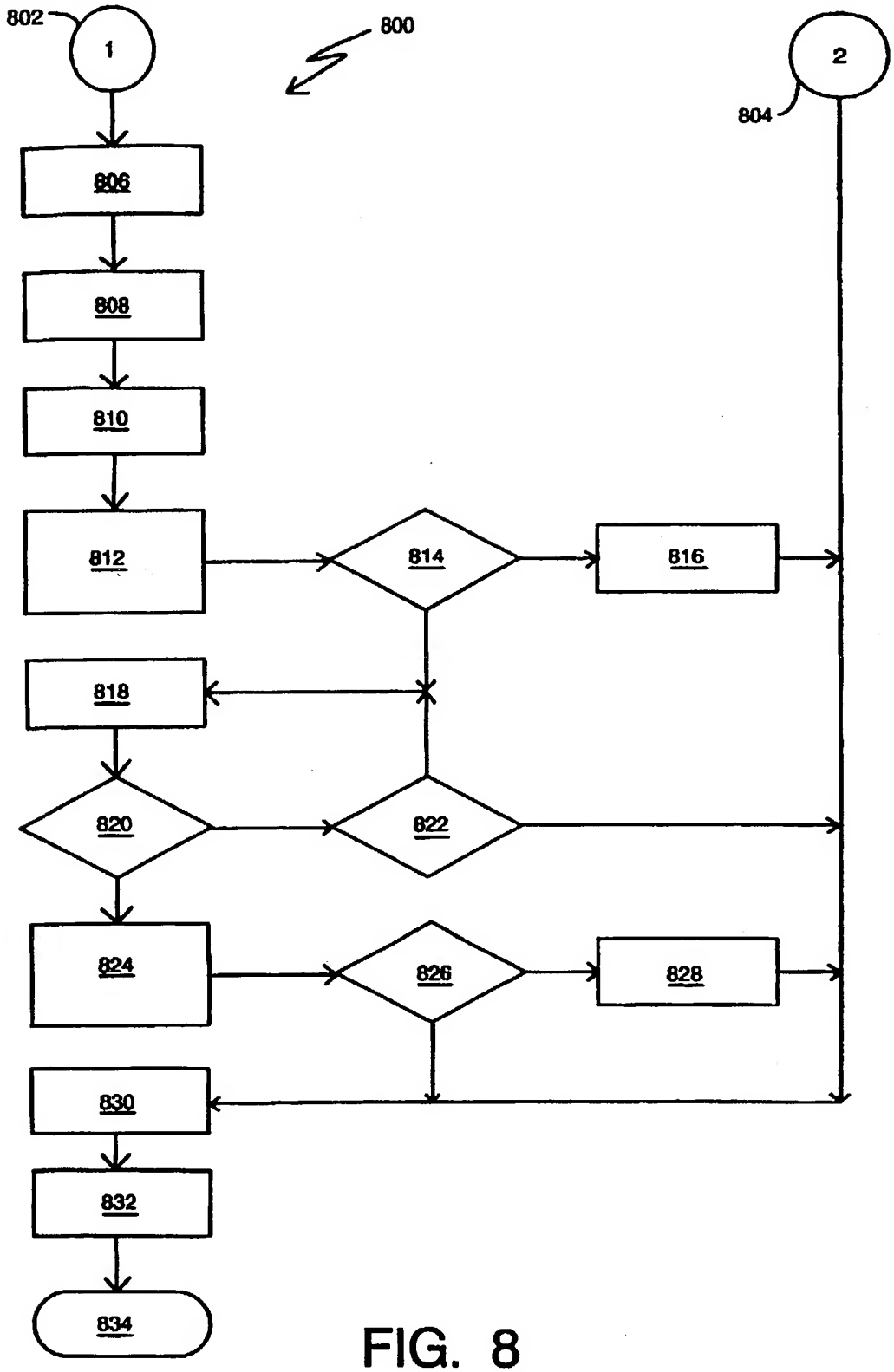

802
1
800
2
804
806
808
810
812
814
816
818
820
822
824
826
828
830
832
834

FIG. 8

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 97/00724

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00 G06F9/445

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 96 15485 A (ABSOLUTE SOFTWARE CORP ;COTICHINI CHRISTIAN (CA); CAIN FRASER (CA)) 23 May 1996	1-4,6-9, 16,17
A	see page 2, line 14 - page 3, line 25 see page 5, line 16 - page 6, line 9 see page 12, line 12 - page 14, line 4; figure 1	5,10,11, 18

X	WO 92 09160 A (TAU SYSTEMS CORP) 29 May 1992	1,6,11, 16-18
A	see page 6, line 2 - page 9, line 6; claims 1,2,4,27-31; figures 1-4	2-5, 7-10, 12-15

A	EP 0 703 531 A (COMPAQ COMPUTER CORP) 27 March 1996 see page 3, line 15-51; figure 1	11,18

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
26 May 1997	05.06.97
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax (+ 31-70) 340-3016	Authorized officer Moens, R

Form PCT ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. ional Application No
PCT/US 97/00724

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9615485 A	23-05-96	AU 3837595 A	06-06-96
WO 9209160 A	29-05-92	US 5103476 A	07-04-92
		US 5222134 A	22-06-93
		CA 2095723 A	08-05-92
		EP 0556305 A	25-08-93
		JP 7089345 B	27-09-95
		JP 6501120 T	27-01-94
EP 0703531 A	27-03-96	US 5586304 A	17-12-96
		AU 3053895 A	21-03-96
		CA 2157728 A	09-03-96
		JP 8227355 A	03-09-96
		US 5588143 A	24-12-96